

How to convert PDF to text from URL asynchronously for PDF to text API in C# and PDF.co Web API

Learn how to convert PDF to text from URL asynchronously to have PDF to text API in C#

The documentation is designed to help you to implement the features on your side. PDF to text API in C# can be implemented with PDF.co Web API. PDF.co Web API is the flexible Web API that includes full set of functions from e-signature requests to data extraction, OCR, images recognition, pdf splitting and pdf splitting. Can also generate barcodes and read barcodes from images, scans and pdf.

C# code snippet like this for PDF.co Web API works best when you need to quickly implement PDF to text API in your C# application. For implimentation of this functionality, please copy and paste code below into your app using code editor. Then compile and run your app. Writing C# application typically includes multiple stages of the software development so even if the functionality works please test it with your data and the production environment.

Our website provides free trial version of PDF.co Web API that includes source code samples to help with your C# project.

C# - Program.cs

```
using System;
using System.IO;
using System.Net;
using System.Threading;
using Newtonsoft.Json.Linq;

// Cloud API asynchronous "PDF To Text" job example.
// Allows to avoid timeout errors when processing huge or scanned PDF documents.

namespace ByteScoutWebApiExample
{
    class Program
    {
        // The authentication key (API Key).
        // Get your own by registering at
        https://app.pdf.co/documentation/api
        const String API_KEY = "*****";

        // Direct URL of source PDF file.
        const string SourceFileUrl = "https://bytescout-
com.s3.amazonaws.com/files/demo-files/cloud-api/pdf-to-text/long-processed-
document.pdf";

        // Comma-separated list of page indices (or ranges) to process. Leave
        empty for all pages. Example: '0,2-5,7-'.
    }
}
```

```

const string Pages = "";
// PDF document password. Leave empty for unprotected documents.
const string Password = "";
// Destination TXT file name
const string DestinationFile = @".\result.txt";
// (!) Make asynchronous job
const bool Async = true;

static void Main(string[] args)
{
    // Create standard .NET web client instance
    WebClient webClient = new WebClient();

    // Set API Key
    webClient.Headers.Add("x-api-key", API_KEY);

    // Prepare URL for `PDF To Text` API call
    string query = Uri.EscapeUriString(string.Format(
        "https://api.pdf.co/v1/pdf/convert/to/text?name={0}&password={1}&pages={2}&url={3}&async={4}",
        Path.GetFileName(DestinationFile),
        Password,
        Pages,
        SourceFileUrl,
        Async));

    try
    {
        // Execute request
        string response = webClient.DownloadString(query);

        // Parse JSON response
        JObject json = JObject.Parse(response);

        if (json["error"].ToObject() == false)
        {
            // Asynchronous job ID
            string jobId = json["jobId"].ToString();
            // URL of generated TXT file that will be
            // available after the job completion
            string resultFileUrl =
                json["url"].ToString();

            // Check the job status in a loop.
            // If you don't want to pause the main thread
            // you can rework the code
            // to use a separate thread for the status
            // checking and completion.
            do
            {
                string status =
                    CheckJobStatus(jobId); // Possible statuses: "working", "failed", "aborted",
                    "success".

                // Display timestamp and status (for
                // demo purposes)
                Console.WriteLine(DateTime.Now.ToLongTimeString() + ": " + status);
            } while (status != "success");
        }
    }
}

```

```

        if (status == "success")
        {
            // Download TXT file
webClient.DownloadFile(resultFileUrl, DestinationFile);

            Console.WriteLine("Generated
TXT file saved as \"{0}\" file.", DestinationFile);
            break;
        }
        else if (status == "working")
        {
            // Pause for a few seconds
            Thread.Sleep(3000);
        }
        else
        {
            Console.WriteLine(status);
            break;
        }
    }
    while (true);
}
else
{
Console.WriteLine(json["message"].ToString());
}
catch (WebException e)
{
    Console.WriteLine(e.ToString());
}
webClient.Dispose();

Console.WriteLine();
Console.WriteLine("Press any key...");
Console.ReadKey();
}


static string CheckJobStatus(string jobId)
{
    using (WebClient webClient = new WebClient())
    {
        // Set API Key
        webClient.Headers.Add("x-api-key", API_KEY);

        string url = "https://api.pdf.co/v1/job/check?jobid="
+ jobId;


        string response = webClient.DownloadString(url);
        JObject json = JObject.Parse(response);

        return Convert.ToString(json["status"]);
    }
}
}
}
}

```



C# - packages.config



FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about PDF.co Web API](#)

[Explore documentation](#)

[Visit \[www.ByteScout.com\]\(http://www.ByteScout.com\)](#)

or

[Get Your Free API Key for \[www.PDF.co\]\(http://www.PDF.co\) Web API](#)